UNIVERSITÀ
DEGLI STUDI
FIRENZE

Mid-term programming assignments

Parallel Programming

What to do

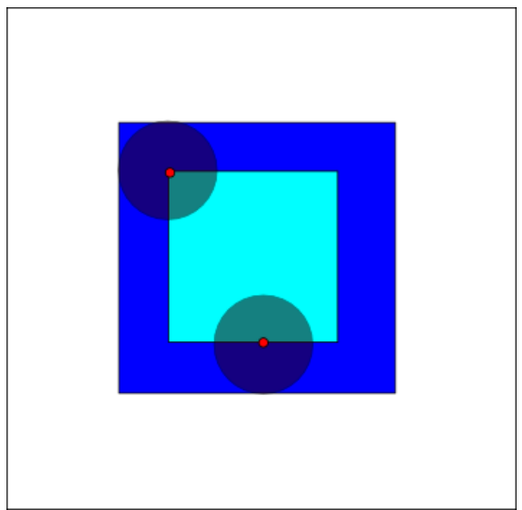
- It’s possible to implement the assignments in pairs
- implement a programming assignment using one of the approaches presented (parallelization/ vectorization). Implement a sequential and a parallel version.
- Write a tech report with performance analysis (speedup) and prepare a presentation
- Keep everything on a public Github/Bitbucket/Gitlab

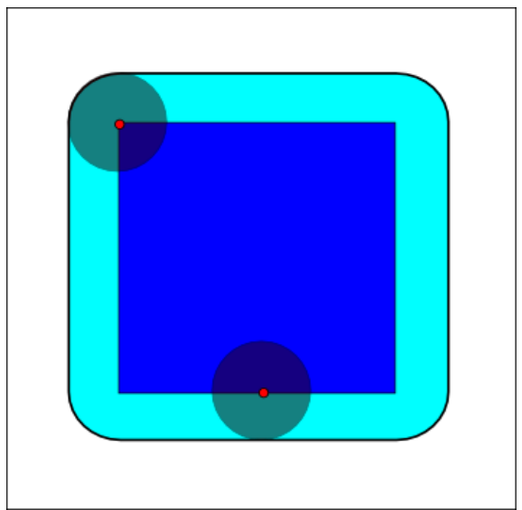
Image reader

- Multithread JPEG image reader
  - read X images from a directory
  - Keep the uncompressed bitmaps (or the compressed stream) in an appropriate data structure, that manages access to the images
  - OpenMP does not allow to implement a non-blocking version since it requires asynchronous multi-threading (ie.. CUDA allows it).

Morphological image processing

- Image processing
  - CUDA is a good candidate
  - See https://en.wikipedia.org/wiki/Mathematical_morphology





Bigrams / trigrams

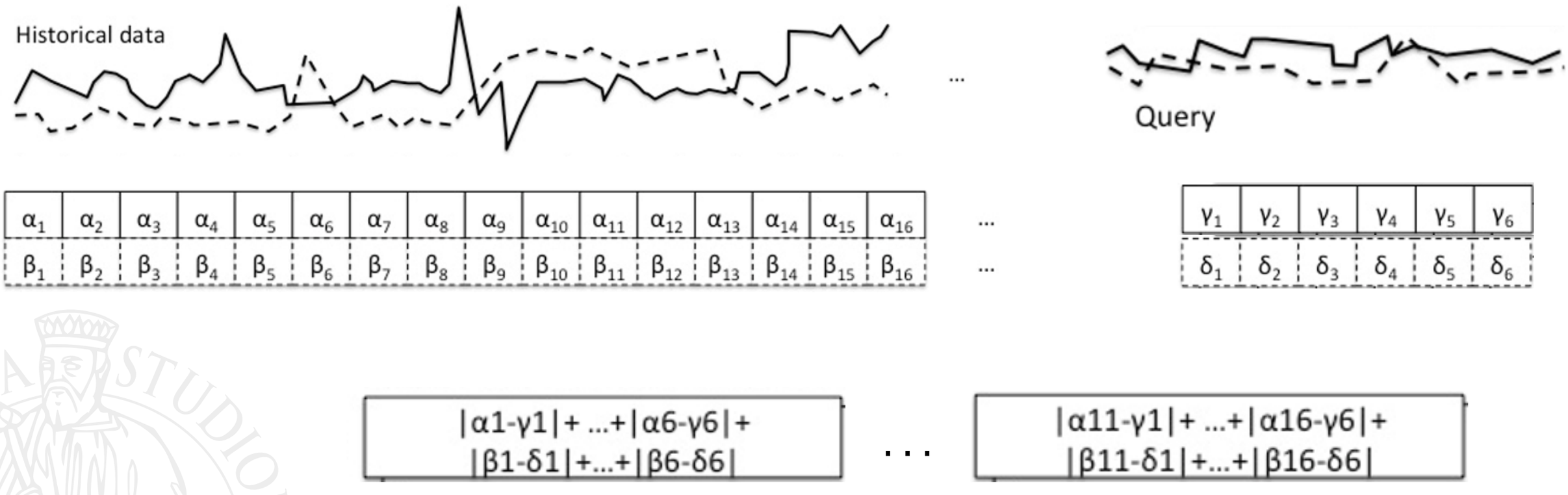
- Compute histograms of bigrams/trigrams on texts (eg. Wikipedia / Gutenberg project documents)
  - Consider bigrams/trigrams of characters and words
  - If needed re-use more and more times the collected data to create a large enough corpus

Password decryption

- Decrypt passwords encrypted using crypt (DES)
  - Consider 8 characters lengths in the set [a-zA-Z0-9./]
  - CUDA does not provide the crypt C-library function. Must use a special implementation (check Moodle)
  - It’s a (slowed down) search problem.
  - Can choose to attack a specific password in a list (in different positions), or decrypt a full list of passwords (in this case reduce the search space)
  - Can consider only dates or common 8-chars passwords

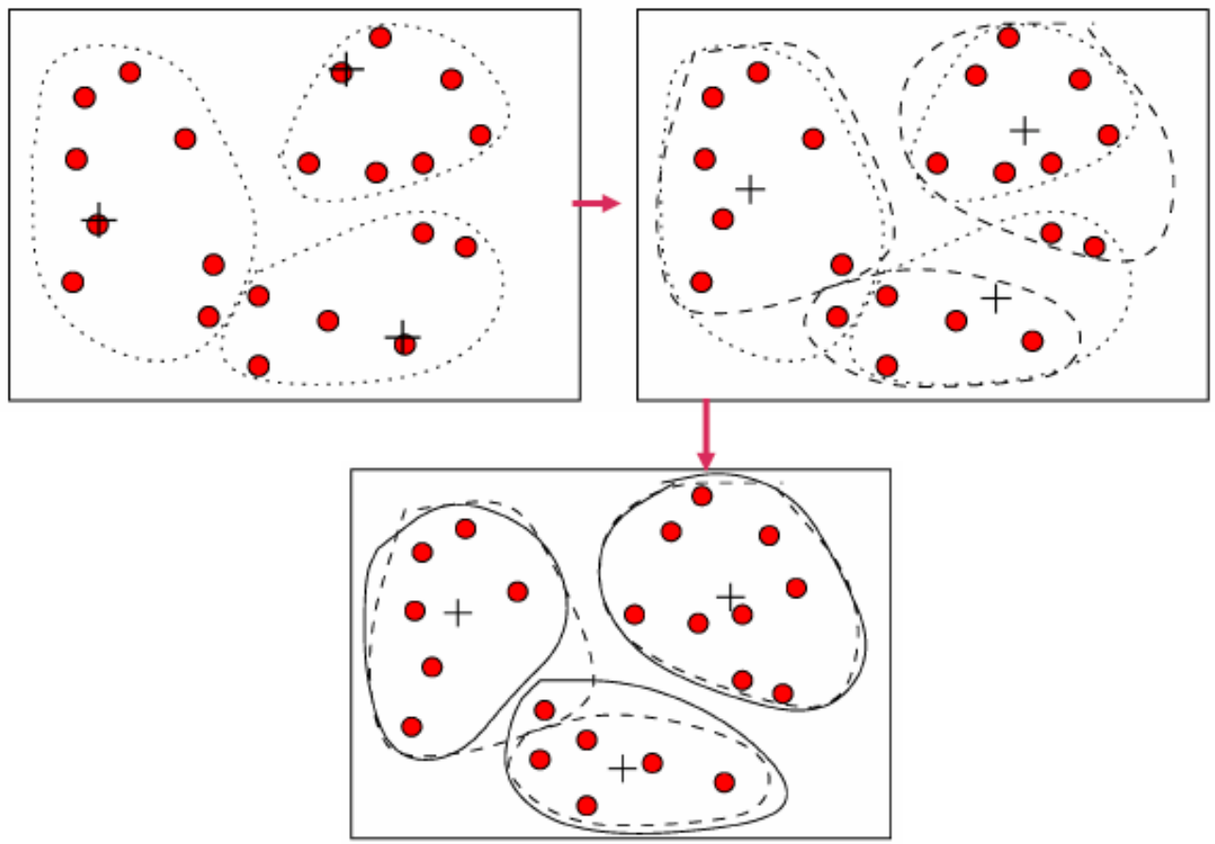
Pattern recognition

- Search a given time series within a larger and longer set of time series
  - Get time series from existing machine learning datasets
- Use SAD as the metric to evaluate the match



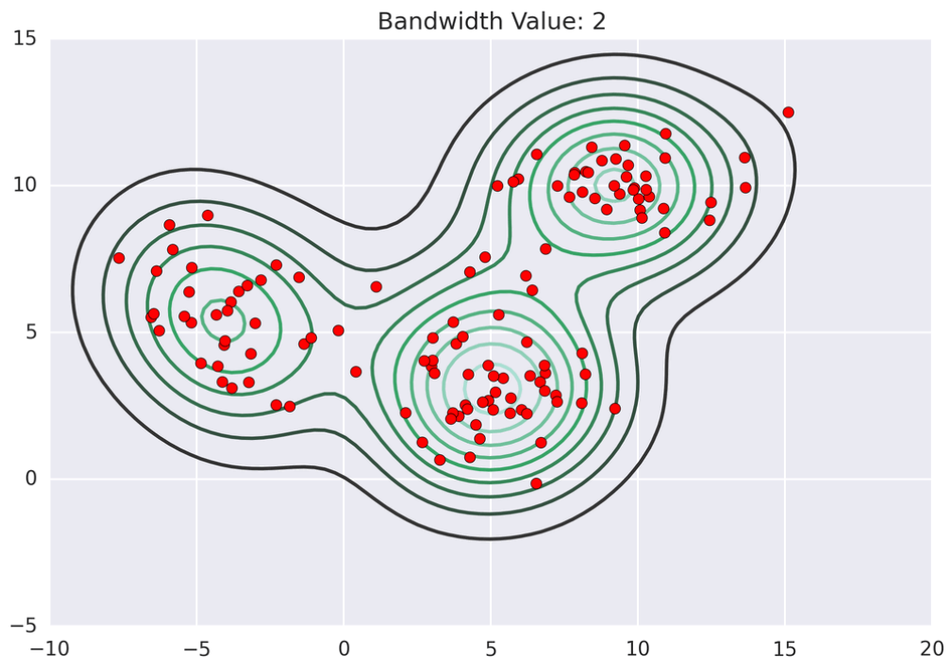
k-means

- Implement a parallel version of k-means clustering
  - There’s no need to implement K-means++ centroid assignments
  - Feel free to chose to operate on 2D or larger dimensionality vectors
- **Esempio**



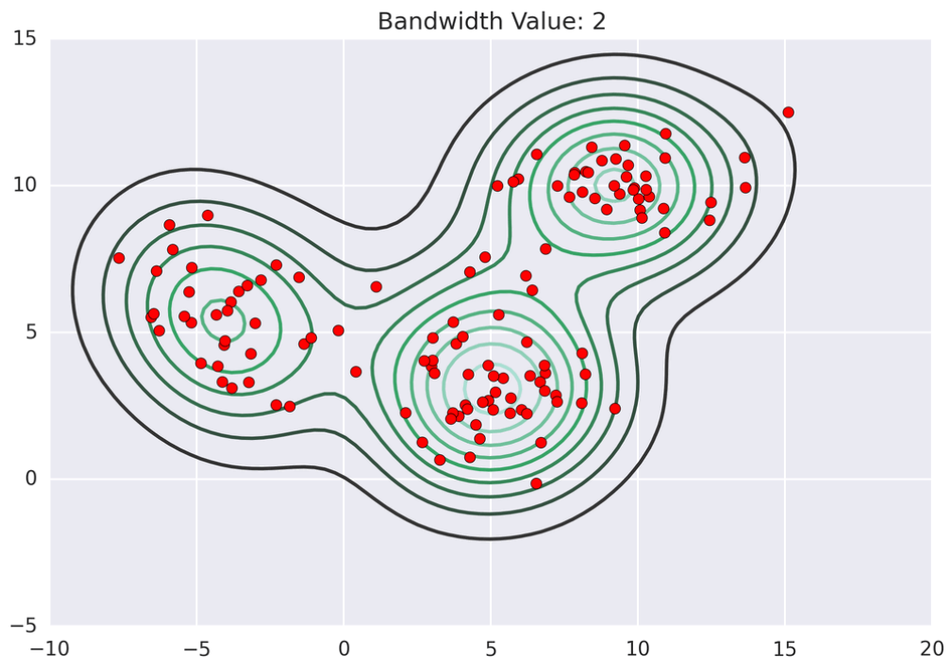
Mean shift clustering

- Implement a parallel version of mean-shift clustering. This clustering doesn't require to specify the number of desired clusters (it uses KDE, so needs a bandwidth parameter) but its complexity is $O(N^2)$
- Can be used to segment images



Mean shift clustering

- Implement a parallel version of mean-shift clustering. This clustering doesn't require to specify the number of desired clusters (it uses KDE, so needs a bandwidth parameter) but its complexity is $O(N^2)$
- Can be used to segment images

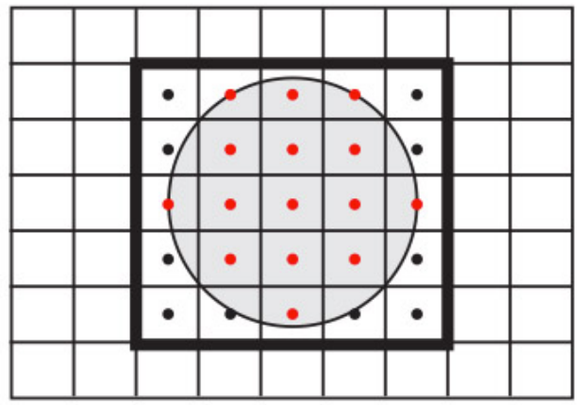


ANN retrieval

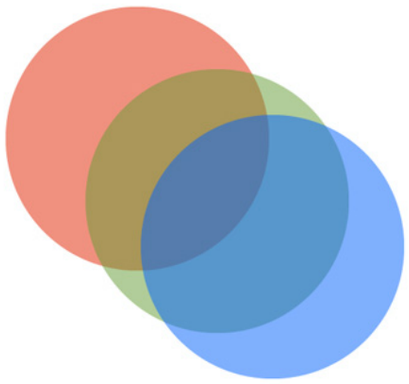
- Consider high dimensionality vectors (eg. 128)
- Given a specified query find the *k nearest neighbors* in a database of vectors
- Can use LSH for approximate indices, eg. using mlpack o LSHkit
- Can get sample datasets from http://corpus-texmex.irisa.fr

Renderer

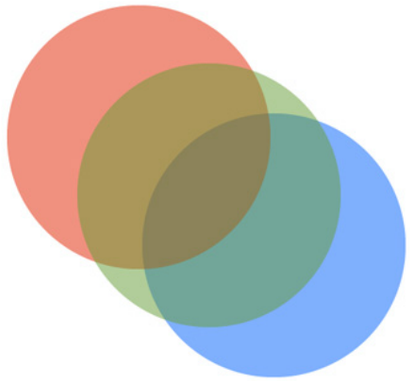
- Implement a renderer that draws (semi) transparent circles (or other shapes). Circles have 3D coordinates and the order along Z axis matters for the correct rendering



- A pixel belongs to a circle if its center is within the circle.



Correct order:
blue over green over red



Incorrect order

Random 10K

Random 100K

Image composition

- Implement a parallel alpha composition of images
- Useful to create image augmentation to train Convolutional Neural Networks, e.g. for detection (see Blend augmentation in IMgAug library)
- Can parallelize applying the same object to different images, or in different positions in the same image

Random maze solver

- Find the exit from a maze using the random movement of a particle.
- Start a large number of particles, move them randomly bouncing on the walls
- Backtrack the first particle to get out of the maze to find the exit